

Servicios WEB: Distribución e integración

Liliana M. Arboleda C.

Universidad Icesi
larboled@icesi.edu.co

Fecha de recepción: 17-06-2004

Fecha de aceptación: 11-10-2004

ABSTRACT

This article presents an overview of web services and the way enterprises should monitor and administer them.

To accomplish this goal, the different standards involved in their development are presented, along with the description of their basic functional framework, emphasizing on how the use of this kind of services provides both a basic process distribution and, an integration of the information managed by the enterprise in its business applications.

Finally, a brief explanation is given about the global benefits accomplished by using web services and the

main implications of their use in an organizational environment.

KEY WORDS

web services, distributed systems, web, web applications, XML, WSDL, SOAP, UDDI.

RESUMEN

En este artículo se presenta un panorama general de los servicios web y la forma en la cual las empresas deben administrarlos y monitorearlos.

Para lo anterior se muestran los diferentes estándares involucrados en su desarrollo, junto con la descripción de su esquema de funcionamiento básico, y se hace énfasis en cómo el uso de este tipo de servicios

proporciona básicamente una distribución de los procesos y a la vez, una integración de la información manejada por la organización en sus aplicaciones de negocios.

Por último, se presentan los beneficios globales logrados por el uso de los servicios web y las principales

implicaciones de su uso en un entorno empresarial.

PALABRAS CLAVES

Servicios web, sistemas distribuidos, web, aplicaciones web, XML, WSDL, SOAP, UDDI

Clasificación Colciencias: B

INTRODUCCIÓN

Entre los principales beneficios que se exponen al hablar de los servicios web, generalmente se encuentran aquellos que tienen que ver con su **granularidad** e **interoperabilidad**, es decir, con la posibilidad de desarrollar componentes de software totalmente independientes que tienen funcionalidad propia, pero que son capaces de exponer tal funcionalidad y compartirla con otros servicios y aplicaciones para lograr crear sistemas más complejos.

Desde este punto de vista, los servicios web son calificados como una nueva etapa del desarrollo de los sistemas distribuidos, que permite aprovechar todas las ventajas de trabajar en ambiente web, unidas a las de contar con una amplia gama de tecnologías que pueden ser utilizadas para el desarrollo de los componentes finales.

Por otro lado, la visión planteada por este paradigma computacional, donde “todo es un servicio”, permite manejar un esquema de **integración universal** en el cual se pueden aprovechar todos los beneficios de cada componente con un nuevo nivel de complejidad y dinamismo.

Los dos planteamientos anteriores han hecho que para muchos no sea muy claro cómo un mismo producto puede proporcionar a la vez **distribución** e **integridad**, como tampoco son muy claros los conceptos relacionados con el desarrollo e implementación de los servicios web.

Este artículo presenta un panorama general de los servicios web, discutiendo:

- a) Los principales estándares utilizados para su desarrollo.
- b) Los beneficios que proporciona su utilización en una organización.
- c) Los retos planteados por la utilización de los servicios web, especialmente en cuanto a su seguridad y administración.

¿QUÉ SON SERVICIOS WEB?

Lo primero que debe tenerse claro es la definición de lo que son los servicios web.

Los servicios web son aplicaciones auto-contenidas, auto-descriptivas y modulares, que pueden ser publicadas, localizadas e invocadas a través del web y que cuentan con un mecanismo estándar para establecer la comunicación con otros tipos de software a través de la red.

“La idea es que cualquier cosa puede ser un servicio web”. “Los servicios web son esencialmente un nivel de infraestructura existente entre modelos de componentes”.¹

Desde el punto de vista de los negocios, los servicios web permiten que las organizaciones integren sus diferentes aplicaciones de una manera eficiente, sin preocuparse por cómo fueron construidas, dónde residen, sobre qué sistema operativo se ejecutan o cómo acceder a ellas. Precisamente por esta razón es que los servicios web se consideran integrado-

1. Timothy Blacke -»Web Service are real«- Oracle Magazine (March/April 2000).

res, porque permiten crear una interfaz de acceso a las aplicaciones, sin importar las características de implementación de éstas.

Previamente, para hacer este tipo de integración, era necesario crear módulos de software especializados, pero actualmente con los servicios web, básicamente se trabaja con estructuras basadas en XML.

Un servicio web simple está caracterizado por cuatro estándares: XML, SOAP, UDDI y WSDL, los cuales al trabajar juntos proporcionan una funcionalidad básica de tipo "solicitud/respuesta". Los servicios web simples pueden utilizarse para entregar de forma eficiente información como noticias, inventarios y reportes de clima a los sitios web, mientras que un servicio web complejo puede involucrar transacciones más elaboradas entre varias partes, involucrando socios de negocios o proveedores y basado en los estándares de e-bussines (por ej.:ebXML). Si un comerciante necesita surtir su tienda, puede utilizar un servicio web para solicitar cotizaciones a través de internet. A su vez, los proveedores podrían responder automáticamente a la solicitud, enviar su oferta a la tienda en línea y la aplicación que origina el mensaje podría seleccionar automáticamente la oferta con una mejor relación costo/beneficio.

FACTORES DE EVOLUCIÓN DE LOS SERVICIOS WEB EN LAS ORGANIZACIONES

En términos generales, la evolución del uso de servicios web en las organizaciones está fuertemente ligada al desarrollo de Internet como red prestadora de servicios. Entre los facto-

res que han impulsado el uso de los servicios web se encuentran:

- *El contenido se está volviendo más dinámico:* Los sitios web actuales proporcionan contenido "instantáneo". Un servicio web debe ser capaz de combinar contenido proveniente de fuentes muy diferentes.
- *El ancho de banda es menos costoso:* Actualmente un servicio web puede entregar tipos variables de contenido, como video o audio. A medida que crezca el ancho de banda, los servicios web deben adaptarse a nuevos tipos de contenido.
- *El almacenamiento es más barato:* Un servicio web debe ser capaz de manejar cantidades masivas de datos, y debe poder hacerlo de forma inteligente.
- *La computación extendida se está volviendo más importante:* Con cientos de millones de dispositivos como teléfonos móviles, beepers, y agendas computarizadas existentes actualmente, estamos llegando a un momento en el cual el PC está dejando de ser el dispositivo más común en internet. A medida que las plataformas se hacen más diversas, tecnologías como XML se volverán más importantes. Un servicio web no puede exigir que los usuarios ejecuten, por ejemplo, un navegador web tradicional en alguna versión de Windows; por el contrario, los servicios web deben servir a todo tipo de dispositivos, plataformas y navegadores, entregando contenido sobre una amplia variedad de tipos de conexión.

Estos factores, unidos a los beneficios proporcionados por los servicios web

en la organización y a los buenos productos disponibles para su desarrollo, han hecho que su utilización se extienda sin mayores obstáculos.

En términos generales, cuando se empiezan a utilizar servicios web en una organización, estos se desarrollan e implementan como servicios simples, que poco a poco se van integrando hasta llegar a servicios web mucho más complejos.

EL PAPEL DE LOS SERVICIOS WEB

Desde un nivel superior, un servicio web se puede definir como una unidad de código que puede ser activada utilizando solicitudes http. Históricamente hablando, el acceso remoto a unidades binarias requiere protocolos específicos de cada plataforma (y algunas veces específicos de cada lenguaje). Ejemplos clásicos de esta situación, son arquitecturas como DCOM, CORBA o RMI: los clientes DCOM pueden acceder a tipos COM remotos utilizando llamadas RPC fuertemente acopladas, CORBA también requiere el uso de un protocolo fuertemente acoplado para activar los tipos remotos y RMI requiere de un protocolo y un lenguaje específicos (Java). El problema con cada una de estas arquitecturas distribuidas (remotas), es que utilizan protocolos propietarios, que generalmente requieren una conexión estrecha a la fuente remota.

Utilizando servicios web, un programador puede emplear cualquier lenguaje que desee, y un consumidor de servicios web puede usar http estándar para invocar métodos en los ti-

pos definidos en el servicio web. Con lo anterior, podría afirmarse que finalmente se logra una integración real de lenguaje y plataforma. Ya no se trata de Java, COM o CORBA, sino de http y el lenguaje de programación que se elija.

La única restricción real a la que debe ponerse atención es que debido a que los servicios web son diseñados para facilitar las invocaciones remotas, debe evitarse la utilización de cualquier lógica basada en la interfaz gráfica de usuario (GUI). Los servicios web generalmente definen objetos de negocio que ejecutan una unidad de trabajo (por ejemplo: realizar un cálculo, leer una fuente de datos, etc.) para el consumidor y esperan la siguiente solicitud.

CARACTERÍSTICAS Y OPERACIONES DE LOS SERVICIOS WEB

El esquema de funcionamiento de los servicios web, requiere de tres elementos fundamentales:

1. Un **proveedor del servicio web**, que es quien lo diseña, desarrolla e implementa y lo pone disponible para su uso, ya sea dentro de la misma organización o en público.
2. Un **consumidor del servicio**, que es quien accede al componente para utilizar los servicios que éste presta.
3. Un **agente de servicio**, que sirve como enlace entre proveedor y consumidor para efectos de publicación, búsqueda y localización del servicio.

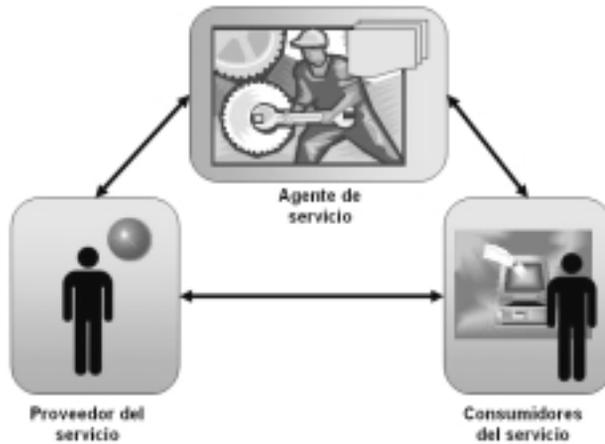


Figura 1. Elementos que interactúan en los servicios web

Al definir un servicio web, se requiere contar con lo siguiente:

- a) Para colocar “disponible” el servicio:
 - El primer paso es **definir** el servicio web: como se dijo antes, el componente software puede desarrollarse utilizando diferentes lenguajes y tecnologías, pero para que este componente pueda interactuar con otros, es necesario utilizar **un lenguaje** común que permita estructurar los datos que componen el servicio web. En este caso tal lenguaje es **XML** (eXtended Markup Language).
 - Una vez se ha definido la funcionalidad del servicio, es necesario **publicarlo** para que otros servicios y aplicaciones puedan acceder a él. Las operaciones de publicación involucran el anuncio del servicio como tal, lo cual corresponde a la ubicación del servicio en un servidor específico y el uso de un servicio de descripción.

El **servicio de descripción** se utiliza para que los clientes puedan saber qué funciones tiene disponibles el servicio web y qué información debe pasarse a esas funciones para poder utilizarlas.

Para que el consumidor pueda utilizar el servicio web, debe entender totalmente los miembros expuestos. Por ejemplo, el consumidor debe saber que el servicio ofrece un método llamado primer Método (string parámetro1, boolean parámetro2, int parámetro3) que toma tres parámetros de tipo string, boolean e int respectivamente, y que devuelve un tipo llamado Respuesta, antes de poder invocarlo. El lenguaje utilizado para elaborar esta descripción del servicio web se denomina **WSDL** (Web Service Description Language) y la publicación se hace mediante UDDI (Universal Description, Discovery and Integration) ya sea en un servidor UDDI público o en uno privado.

b) Para “localizar” el servicio:

- Cuando un consumidor desea acceder a un servicio web, debe contar con un **servicio de descubrimiento**, que permita conocer la ubicación exacta del servicio, es decir, se debe contar con un directorio donde se tengan listas las referencias a los servicios disponibles. Esto se logra gracias a UDDI.

Para intercambiar información hacia y desde el servicio (entre los consumidores y los servicios web), es necesario contar con un **protocolo de comunicaciones** como SOAP, el cual trasmite los datos sobre HTTP, utilizando métodos como GET o POST. La información entre un servicio web y su consumidor se transmite en formato XML.

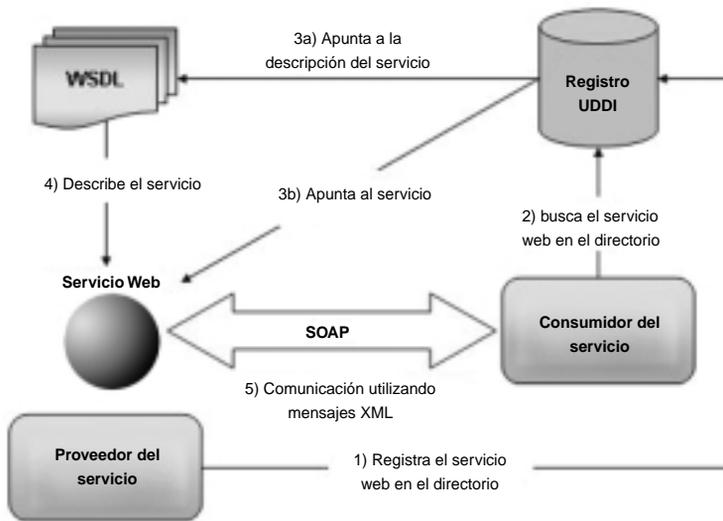


Figura 2. Publicación y descubrimiento de los servicios web

Estándares para la caracterización de los servicios web

Pila de interoperabilidad de los servicios web	UDDI (Universal Description, Discovery and Integration)
	WSDL (Web Services Definition Language)
	SOAP (Simple Object Access Protocol)
	XML (eXtensible Markup Language)
	HTTP, TCP/IP (Protocolos comunes de Internet)

Figura 3. Pila de interoperabilidad de los servicios web

- **XML: eXtensible Markup Language**

La definición dada por el World Wide Web Consortium (los responsables de XML) es:

«XML describe una clase de objetos de datos llamados documentos XML, y describe parcialmente el comportamiento de los programas que los procesan. Los documentos XML están formados por unidades de almacenamiento llamadas entidades, las cuales contienen datos analizados (parsed) o sin analizar (unparsed). Los datos analizados están formados por caracteres, algunos de los cuales conforman datos y otros etiquetas.»

Aunque esta definición es un poco compleja por la cantidad de elementos involucrados, es importante resaltar que XML permite representar documentos («documento» se refiere no sólo a textos, sino también gráficos, correos electrónicos, transacciones de comercio electrónico, etc.) y que todo documento XML posee una

estructura lógica y una física. Físicamente, el documento está compuesto de unidades llamadas entidades. Una entidad, en términos más simples, puede verse como un «trozo de documento», es decir, puede ser un simple carácter o todo el capítulo de un libro. Todo documento tiene una entidad raíz.

En un documento XML es posible hacer referencia a entidades internas o externas, indicando que están definidas en el mismo documento o en otros documentos (que pueden estar en otros archivos del disco duro, en una base de datos, en internet, etc.), lo que hace muy flexible la construcción de documentos XML.

Lógicamente el documento XML está formado por elementos, que representan la estructura de la información que se está manejando, por ejemplo, si se fuera a representar un libro puede definirse que está formado por los siguientes elementos (Ver Figura 4):

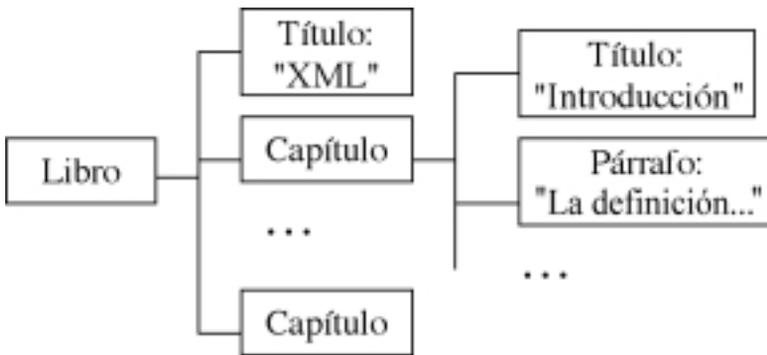


Figura 4. Representación de la información de un libro en XML

Cada elemento puede constar de otros elementos, así como de frases y palabras (normalmente el contenido del documento). El elemento que contiene a los demás (en el ejemplo, libro), se denomina elemento raíz, indicando que no depende de otros elementos.

Para definir un elemento en XML se tienen etiquetas y contenido, que en XML se denominan datos de tipo carácter.

Las estructuras física y lógica de un documento se organizan adecuadamente, para lo cual se han establecido unas normas que definen cuándo un documento XML está bien formado y cuándo es válido, o no.

- **WSDL: Web Services Definition Language**

WSDL es el lenguaje común utilizado para la descripción de los servicios web. Es un lenguaje basado en XML que describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios web existentes en una máquina dada, los métodos que soportan y la sintaxis de los protocolos de comunicación (http, SOAP).

En términos generales, un documento WSDL contiene información acerca de la interfaz, la semántica y los aspectos administrativos involucrados en una solicitud (llamado) a un servicio web.

Según el consorcio W3C (World Wide Web Consortium), encargado de la definición del estándar, WSDL es *«un formato XML para la descripción de servicios de red, como un conjunto de puntos finales operando sobre mensajes que contienen información orientada a documentos u orientada a*

procesos». Las operaciones y mensajes se describen en forma abstracta y luego se asocian a un protocolo de red o a un formato de mensaje específico, para definir un punto final. WSDL se extiende para permitir la descripción de los puntos finales y sus mensajes, sin importar qué formatos de mensaje o protocolos de red se estén utilizando para la comunicación.

- **SOAP: Simple Object Access Protocol**

El protocolo SOAP (Simple Object Access Protocol) utiliza mensajes XML para invocar métodos remotos. Un servicio web podría interactuar con servicios remotos a través de los métodos get y post de http, pero SOAP es mucho más robusto y flexible.

SOAP es un protocolo liviano, basado en XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Sin embargo, SOAP no define la aplicación, ni la semántica de implementación. En vez de esto, proporciona un modelo de empaquetamiento modular y los mecanismos para la codificación de los datos dentro de los módulos. Esto permite que el protocolo simple se utilice en una amplia variedad de sistemas modulares y que cumpla su propósito primario de facilitar la interoperabilidad entre componentes de software heterogéneos.

SOAP es una especificación para la invocación de métodos en servidores, servicios, componentes y objetos, y codifica la práctica existente de utilizar XML y http como un mecanismo de invocación de métodos.

Un paquete SOAP contiene información que se puede utilizar para invo-

car un método. La especificación SOAP no define la forma de llamar al método; tampoco maneja recolección de basura distribuida, ni la seguridad de tipos o http bidireccional. Lo que SOAP permite es el paso de parámetros y comandos entre clientes y servidores de http, independientemente de las plataformas y aplicaciones existentes en el cliente y en el servidor. Los parámetros y los comandos se codifican utilizando XML.

• **UDDI: Universal Discovery, description and integration**

Para los servicios web, XML es el mecanismo por medio del cual los objetos de negocio acuerdan lo que van a hablar unos a otros, SOAP indica cómo van a efectuar la comunicación y UDDI les permite conocer con quién se van a comunicar y dónde encontrar otros objetos de negocio.

En pocas palabras, UDDI es una especificación para un registro distribuido de información acerca de los servicios web. Define la forma en la cual se publica y descubre información acerca de éstos.

Un registro basado en UDDI es donde se pueden descubrir los servicios web. El método utilizado por UDDI para el descubrimiento de servicios es tener un registro de aquellos servicios que se encuentran distribuidos a través del web. En el registro distribuido, los negocios y los servicios se describen utilizando un formato XML común. Los datos estructurados en esos documentos XML son de fácil búsqueda, análisis y manipulación.

Debe anotarse que UDDI/SOAP no son los únicos modelos para el descubrimiento y manejo de mensajes en los servicios web. EbXML ha sido de-

sarrollado para un fin similar, así como para proporcionar una interfaz de negocio, seguridad robusta y otras funciones que permiten el manejo de transacciones reales de e-business. ebXML y UDDI/SOAP son tecnologías complementarias.

Como su nombre lo indica, el estándar de descripción universal, descubrimiento e integración (UDDI), proporciona un mecanismo para que los objetos de negocio se describan a sí mismos y a los tipos de servicios que proporcionan, para luego registrarse y publicarse en un registro UDDI. Tales negocios “públicos” pueden ser buscados, consultados o descubiertos por otros objetos de negocio, utilizando mensajes SOAP. Después de descubrir otros negocios similares con los cuales pueden asociarse, los objetos de negocio pueden utilizar este mecanismo para “integrar” sus servicios con sus “socios” y proporcionar servicios a sus clientes.

Cuando se publica un servicio, es necesario definir al menos cuatro estructuras de información, que son:

a) **Entidad de Negocio:** Esta estructura captura información acerca de un negocio o entidad y es utilizada por los objetos de negocio para publicar información descriptiva acerca de sí mismos y de los servicios que ofrecen.

Las descripciones de servicio y la información técnica se expresan dentro de una Entidad de Negocio.

b) **Servicio de Negocio:** Esta estructura representa los servicios o procesos de negocio proporcionados por la entidad de negocio. Generalmente contiene la clave única utilizada para representar el

servicio, su nombre “human-readable”, una descripción opcional y las estructuras de plantilla de enlace que guardan la información técnica.

c) **Plantilla de Enlace:** Esta estructura representa los datos importantes que describen las características técnicas de la implementación de servicio dada. Cada plantilla tiene una clave de enlace única, la clave de servicio asociada y el punto de acceso que representa la dirección para llamar a un servicio web dado.

d) **ModeloT:** El papel principal de un ModeloT es representar una especificación técnica. Tiene una llave, un nombre, un descriptor opcional y un URL desde el cual se puede obtener información adicional acerca de los datos.

UDDI proporciona dos grandes categorías de APIs, el API de publicación (Publish API) y el API de consulta (Inquire API).

El API de publicación proporciona el mecanismo para que los proveedores de servicio se registren, junto con su servicio, en el registro UDDI. El API de consulta permite que quienes se suscriben al servicio busquen los servicios disponibles. Este API proporciona dos tipos de llamados, un mecanismo de descubrimiento y un mecanismo de obtención, cuando está disponible toda la información requerida para buscar un servicio.

Los mensajes en el API de publicación representan comandos que se utilizan para publicar y actualizar información contenida en un registro UDDI-compatible. Este API consiste de las siguientes funciones:

- Cuatro mensajes para guardar cada una de las cuatro estructuras: `save_business`, `save_service`, `save_binding`, `save_tModel`.
- Cuatro mensajes para borrar cada una de las cuatro estructuras: `delete_business`, `delete_service`, `delete_binding`, `delete_tModel`.

Los mensajes en el API de consulta representan consultas que cualquiera puede hacer a un registro UDDI. Los mensajes se pueden clasificar en dos grupos: de Navegación (Browse) y de Profundización (Drill-down).

BENEFICIOS DE LOS SERVICIOS WEB

1. *Promueven la interoperabilidad:* La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje. Esta interacción requiere un documento WSDL para definir la interfaz y describir el servicio, junto con un protocolo de red (generalmente http).
2. *Permiten la integración “justo-a-tiempo”:* El proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicio utilizan a los agentes para encontrar proveedores de servicio. Una vez el solicitante y el proveedor de servicio se han ubicado, se utiliza el documento WSDL del proveedor para enlazar al solicitante con el servicio. Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son auto-configurables, adaptativos y robustos.
3. *Reducen la complejidad por medio del encapsulamiento:* Los so-

licitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementado el servicio por parte del proveedor, y éste a su vez, no sabe cómo utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores. El encapsulamiento es crucial para reducir la complejidad.

4. *Dan una "nueva vida" a las aplicaciones de legado:* Es relativamente correcto tomar una aplicación, generar un wrapper SOAP, luego generar un documento WSDL para moldear la aplicación como un servicio web.
5. *Abren la puerta a nuevas oportunidades de negocio:* Los servicios web facilitan la interacción con socios de negocios, al poder compartir servicios internos con un alto grado de integración.
6. *Disminuyen el tiempo de desarrollo de las aplicaciones:* Pues gracias a la filosofía de orientación a objetos utilizada, el desarrollo se convierte más bien en una labor de composición.

IMPLEMENTACIÓN DE LOS SERVICIOS WEB EN UNA ORGANIZACIÓN

Generalmente, la implementación de servicios web en las organizaciones empieza por servicios sencillos, que prestan un único servicio a otras aplicaciones o que usualmente están enlazados a una aplicación web específica que pone los resultados del servicio a disposición del usuario final.

Una vez se ha logrado un funcionamiento estable de los servicios indi-

viduales, se inicia el desarrollo de servicios web más complejos, que involucran aspectos como interacción entre servicios y sincronización de sus funciones, pero que potencian las ventajas de estos componentes de software como integradores de los procesos de la compañía.

Esta interacción impulsa la aparición de nuevas variables que deben tenerse en cuenta para lograr el adecuado funcionamiento de las aplicaciones basadas en servicios web distribuidos. Entre estas variables se tienen:

- a) Al tratarse de componentes distribuidos, la complejidad de interacción de los mismos aumenta y deben considerarse aspectos como confiabilidad de la red, latencia, seguridad y disponibilidad. Por ejemplo, si un sistema utiliza tres servicios web diferentes para ejecutar una única transacción y uno de ellos falla, es posible que no pueda prestarse el servicio solicitado por el usuario final.
- b) En cierta forma se pasa de una estructura tipo cliente/servidor (solicitud/respuesta), a una estructura punto a punto, donde las interacciones de un servicio web no se hacen directamente contra librerías locales, sino que se interactúa con recursos remotos controlados por otros servicios web. Los consumidores de los servicios web son otros componentes de software, no personas.
- c) El mantenimiento de los sistemas basados en servicios web también puede ser muy dispendioso, pues en aquellos casos en los cuales los servicios asociados al sistema no sean administrados por una mis-

ma organización, el proceso de arreglar un problema y el costo asociado a ese proceso, serán mucho mayores.

- d) Las modificaciones a cada servicio web deben ser planeadas cuidadosamente, pues debe tenerse en cuenta que un mismo servicio web puede estar siendo utilizado por diferentes aplicaciones, y peor aún, de diversos tipos. Por tratarse de componentes software, estos no pueden “reaccionar instantáneamente” a los cambios del servicio que están utilizando

Lo anterior dificulta la puesta a punto y la administración de los servicios web para garantizar que estos puedan ser utilizados en un ambiente de producción, y por tanto, en la parte administrativa de los servicios web, es necesario tener en cuenta:

1. aspectos relacionados con el desempeño y la disponibilidad del sistema (características operativas)
2. aspectos relacionados con la confiabilidad y el control de acceso al sistema (características de seguridad)
- 3) aspectos relacionados con los beneficios logrados a través del uso del sistema (características de negocio).

CONCLUSIONES

Los servicios web son una realidad dentro de las aplicaciones disponibles para su uso dentro de una empresa o para la interacción entre diferentes unidades de negocios, y su evolución se debe al uso de estándares muy bien definidos como XML, WSDL, SOAP y UDDI, pero es importante tener en cuenta que no sólo se trata de “tener”

servicios web que faciliten la distribución de procesos y la integración de información, sino que es necesario monitorear y controlar cuidadosamente su funcionamiento, para lograr los beneficios deseados.

Para la puesta a punto de los servicios web en una empresa, es necesario afinar parámetros relacionados con:

- a) Lógica del negocio: refinar el código para eliminar condiciones de error inesperadas.
- b) Granularidad: definir cómo será la interacción con el servicio web, si utilizando un esquema solicitud respuesta o mediante paso de mensajes.
- c) Acoplamiento: si se diseña el servicio como un servicio síncrono, el acoplamiento es más sencillo, pero implica una interacción de tipo bloqueante entre el consumidor y el servicio web. Si se diseña con un esquema asíncrono, se disminuye la confiabilidad del sistema, pero se evitan las operaciones bloqueantes.
- d) Interoperabilidad: el servicio web debe manejar los estándares del mercado para garantizar el intercambio de mensajes con otros servicios.

BIBLIOGRAFÍA

1. *Introduction to web services*. Rogue Wave Software. <http://www.roguewave.com>. (2004)
2. *Web services: A practical introduction to SOAP web services*. Systinet Coop. (2003)

3. *Turning software into a service.*
<http://computer.org>. (Octubre 2003)
4. *Practical considerations for implementing web services. The role of web services management.*
AmberPoint (Octubre 2003)
5. *The pervasiveness of web services.*
<http://www.mercuryinteractive.com>
(2003)
6. *Web services are real.*
Oracle magazine (marzo/Abril 2002)

CURRÍCULO

Liliana M. Arboleda C. Ingeniera de Sistemas de la Universidad Icesi. Especialista en Dirección estratégica en Tecnologías de la Información y las Comunicaciones de la Universidad Politécnica de Madrid. Se desempeñó como analista de sistemas en Carpak S.A. y actualmente es profesora-investigadora del departamento de Redes y Comunicaciones de la Universidad Icesi, en campos relacionados con protocolos de telecomunicaciones, servicios de red, sistemas distribuidos y programación orientada a objetos. ☼